RELATED APPLICATIONS

sub a'

~~This patent application is related to co-pending U.S. Patent Application 09/755,874, (Attorney's Docket Numbers: MS1-710US).~~

TECHNICAL FIELD

This invention relates to managed devices, and more particularly to methods and arrangements for providing improved software version control in managed devices.

BACKGROUND

motivation

Managed devices, such as computers, set-top boxes, entertainment centers, communication devices, and the like, often require periodic updates to their software suite. Such updates are typically carried out via another computer or device that is coupled to the managed device and arranged to download the updated software or other data to the managed device. For example, a server computer can be configured to provide downloadable updates to client devices over a communication link.

To avoid downloading too many files, and/or the wrong files during an update or upgrade there is a need for some kind of a software version control mechanism or technique. Preferably, the software version control mechanism or technique will be configured to expedite the overall downloading of files by quickly identifying which, if any, files require updating.

CON (1/4)

carries the arbitrary extension *.cim. The *.cim files are then stored on the FAT file system directly.

Each image preferably contains files that are functionally related. For example, all IE browser files, such as mshtml.dll, shdocvw.dll and WinINET.dll are preferably compressed into one image whose name maybe "a6945jtv451ktc909btes6lmv2.cim".

A search of the regular FAT file system only reveals many *.cim files. Unfortunately, this search result is of little use to regular applications, which do not understand the inter-workings of the compressed *.cim files.

To solve this problem, a new file system mounting port has been developed. When an application accesses the files through the new path, the compressed file system driver is invoked and used to open the *.cim on the FAT drive and read out the file. Since the driver understands the compression scheme used in the *.cim, it will be able the feed the correct data to the application.

With the help of a compressed file system, client device 106 uses a system of unique files to ease the control of software revisions. Here, the client software consists of multiple compressed files and a list file that contains the list of all the names of the compressed files (*.cim). Every time the software is built, the compressed image file name is changed, and the resulting names are always globally unique. Therefore, if two compressed images share the same name, every file inside each of these is identical. In other words, they have the same revision. If the names are different, then the versions are not the same.

This feature makes version comparison and incremental upgrading extremely easy. To decide which files need to be upgraded, client device 106 only needs to check with server device 102 and compare the server's most up-to-date

CON(3/4)

list with its own list of file names. If a file name from the server's list does not show up on the local driver, client device 106 needs to download that particular file. Since the files are preferably grouped by functionality when being compressed into *.cim files, usually only a small portion of *.cim files needs to be download and updated.

Fig. 5 is a flow diagram depicting a method 500 for providing software version control in a client device 106. In step 502, one or more files are determined to make up a group, wherein the software is divided as such into a plurality of groups. In step 504, each of the groups of files is compressed to form corresponding compressed images. Next in step 506, each compressed image is associated with a unique identifier. The unique identifier can be derived from a portion of the compressed image, for example. Then, in step 508, a listing of unique identifiers is generated. In step 510, each of the compressed images is stored with the memory of client device 106. Next, in step 512, during an upgrade process, the listing of unique identifiers within client device 106 is compared with the latest listing of unique identifiers as provided by server device 102. The compressed images that are missing from the client's listing of unique identifiers as determined from the comparison in step 512 are then selectively downloaded in step 514 from server device 102 to client device 106.

Support for claim 21

Although some preferred implementations of the various methods and arrangements of the present invention have been illustrated in the accompanying Drawings and described in the foregoing Detailed Description, it will be understood that the invention is not limited to the exemplary implementations disclosed, but is capable of numerous rearrangements, modifications and

CON(4/4)

listing of unique identifiers but not in the current listing of unique identifiers in the client device". Nether *Stuart, Spanbauer, Suzuki, and/or Hollingsworth* disclose these recited features of claim 1.

Miller only describes generating file names by hashing the entire contents of a package. The Action admits at page 7 of the Action that this is not taught by *Stuart, Spanbauer,* and/or *Suzuki.* To provide this missing teaching, the Office Action relies on *Hollingsworth.* However, *Hollingsworth* is completely silent on deriving CDNs (Content Derived Names) from a "portion" of anything. Instead, Hollingsworth explicitly describes at page 1, section 1, paragraph 4, that "Content Derived Names are computed by hashing the contents of a file using a secure hash". Nowhere does Hollingsworth teach or suggest that "contents of a file" is anything less that all contents of a file. Thus, a system *Stuart, Spanbauer, Suzuki,* and/or *Hollingsworth* may never "deriving a unique identifier of the unique identifiers for the one processed image, the unique identifier being derived as a function a portion of the one processed image, the portion being less than a whole of the one processed image", as claim 1 recites.

Accordingly, withdrawal of the 35 USC §103(a) rejection of claim 1 is respectfully requested.

Claims 2, 4, 5, and 25 depend from claim 1 and recite additional features. At least for reasons of this dependency, claims 2, 4, 5, and 25 are allowable over the cited combination.

Accordingly, withdrawal of the 35 USC §103(a) rejection of claims 2, 4, 5, and 25 is respectfully requested.

Claim 8 recites in part "assigning each of a plurality of data files to one of a plurality of specific corresponding downloadable file groups", "generating

RMK (1/1)

21. (Currently amended) A system comprising:

21-1 a network;

21-2 a server device operatively coupled to the network, the server device being configured to:

21-2-1 assign each of a plurality of server-based data files to one of a plurality of specific corresponding server-based downloadable file groups;

21-2-2 generate processed images and a listing of unique identifiers as follows:

for each server-based downloadable file group, the server device is configured to:

21-2-2-1 compress together data files assigned to the server-based downloadable file group to form one processed image ; and

* 21-2-2-2 derive a unique identifier for the one processed image, the unique identifier being derived based on a portion of the processed image, the portion being less than a whole of the processed image;

21-2-3 selectively output the processed images and a latest listing of the unique identifiers over the network; and

21-2-4 a client device operatively coupled to the network, the client device being configured to communicate with the server device through the network, wherein the client device is further configured to maintain a listing of unique identifiers associated with processed images stored locally within the client device, download the latest listing of unique identifiers, compare the listing of unique identifiers with the latest listing of unique identifiers , and selectively download processed images whose unique identifiers appear in the latest listing of

CLM (1/2)

unique identifiers from the server device but do not appear in the listing of unique identifiers in the client device.

22. (Canceled)

23. (Previously presented) The system as recited in Claim 21, wherein the server device is further configured to selectively assign a plurality of related function data files to one downloadable file group.

24. (Canceled).

25. (Previously presented) The method as recited in Claim 1, wherein the one processed image for the downloadable file group has a ".cim" extension.

26. (Previously presented) The computer-readable medium as recited in Claim 8, wherein the respective processed image for the downloadable file group has a ".cim" extension.

27. (Previously presented) The apparatus as recited in Claim 15, wherein the one processed image for the downloadable file group has a ".cim" extension.

CLM (2/2)